

File



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/537,998	03/29/2000	David D'Souza	MSFT115144	6844

26389 7590 12/01/2004

CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC
1420 FIFTH AVENUE
SUITE 2800
SEATTLE, WA 98101-2347

EXAMINER

LAO, SUE X

ART UNIT	PAPER NUMBER
----------	--------------

2126

DATE MAILED: 12/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/537,998

Applicant(s)

D'SOUZA, DAVID

Examiner

Sue Lao

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 28 June 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-15 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 8-10 and 15 is/are allowed.
- 6) ☒ Claim(s) 1-3, 5-7, 11, 12, 14 is/are rejected.
- 7) ☒ Claim(s) 4 and 13 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892) *
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-15 are pending. This action is in response to the amendment filed 6/28/2004. Applicant has amended claim 12.
2. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.
3. Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Prorise and Stinson in view of Notes (Lecture Notes "Non-Preemptive Scheduling").

It is noted that Prorise and Pietrek discuss features of Windows 3.1 and Stinson is cited to show that Windows 3.1 was available back in 1992 and therefore the teaching of Prorise and Pietrek was available back in 1992.

As to claim 1, Prorise teaches tasks (Windows applications, DOS applications), logically partitioning tasks into groups of interdependent tasks (group of Windows applications to be run in System VM, and group of DOS applications to be run in DOS VMs), preemptively scheduling the groups (preemptive scheduler, divides CPU time among System VM and DOS VMs preemptively), each group is given a time slot (time slicing), non-preemptively scheduling within a group (non-preemptive scheduler associated with Windows applications within System VM), see fig. 1; pg. 261, left col., 2nd para.; pg. 263, right col., section "Windows Does It Too" to pg. 264, left col..

It is noted that Windows OS and DOS each provides its own system routines, ie, resources, to be utilized/called by application programs running thereon. Windows applications grouped under System VM call/utilize Windows system routines / resources, and DOS applications grouped under DOS VMs call/utilize DOS system routines / resources.

Prorise does not explicitly teach that the non-preemptively scheduling is performed within *each* of all the groups, which, however, is met by Notes which teaches both DOS and Windows 3.1 separately employ non-preemptively scheduling. See pages 1-2. Therefore, it would have been obvious to non-preemptively schedule within

Art Unit: 2126

each group in Prosise. One of ordinary skill in the art would have been motivated to combine the teachings of Prosise and Notes because this would have provided reduced scheduling overhead (Notes, page 2, lines 8-10).

It is noted that although the Notes was dated 2000, the technologies it describes, ie, DOS OS and Windows 3.1 OS, was available back in 1992. Therefore the teaching of Notes was available back in 1992.

4. Claims 2, 3, 11 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Prosise and Stinson in view of Notes as applied to claim 1 and further in view of Pietrek.

As to claim 2, Pietrek teaches (pg. 225, 3rd and last para.s) storing a group list for each associated group (task database TDB, maintained in a linked list), identifying information for tasks (selector of next TDB). Prosise and Pietrek both discuss features of Windows 3.1 and thus it would have been obvious to combine the teachings.

As to claim 3, Pietrek teaches (TDB, table 3-2) storing status information (flags, error flags, etc.), whether the group has a task that is running (DLL is loading, DLL is unloading), holding identifying information about any task that is running (module handle for task). It is noted that an application program conventionally comprises some code and therefore running a task/application would involve running a module of code.

As to claim 11, note discussion of claim 1 for partitioning, preemptively scheduling and non-preemptively scheduling. Using a partitioning mechanism to perform partitioning, a preemptive scheduler to perform preemptive scheduling and a non-preemptive scheduler to perform non-preemptive scheduling would have been inherent/obvious. Prosise and Pietrek further teaches execution mechanism for executing the tasks (processor).

As to claim 12, note discussion of claim 1 for preemptively scheduling and non-preemptively scheduling. Prosise and Pietrek further teaches operating system (Windows 3.x, Windows 3.1), logically partitioning tasks into groups of interdependent tasks to be run non-preemptively (group of Windows applications to be run in System VM, and group of DOS applications to be run in DOS VMs), and tasks allocating the

same resources, and tasks that do not allocate the same resources being placed as separate groups in that Windows OS and DOS each provides its own system routines, ie, resources, to be allocated/called by application programs running thereon. Windows applications grouped under System VM allocate/utilize Windows system routines / resources, and DOS applications grouped under DOS VMs allocate/utilize DOS system routines / resources.

5. Claims 5-7, 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Prosise and Stinson in view of Notes as applied to claim 1 and further in view of Pietrek and Tulpule et al.

As to claim 5, it is covered by claim 1 except for (1) modules of code, (2) providing a task dependency list for each task, the task dependency list listing resources that are candidates to be allocated when the task is run, (3) that the logically partitioning includes examining the task dependency lists to group the tasks into groups of interdependent tasks.

As to (1)-(2), Pietrek teaches modules of code (module, code, pg. 213, last para.), providing a task dependency list for each task (task database TDB), the task dependency list listing resources that are candidates to be allocated when the task is run (OpenApplEnv() creates a segment that contains the module handles of all the DLLs of the loading executable module; pg. 252, 1st para.; module handle for task; Table 3-2). Note discussion of claim 2 for a motivation to combine.

As to (3), Prosise teaches interdependent tasks (windows applications) are grouped together (grouped to be run under System VM), wherein the modules run by the tasks (windows resources, windows DLLs) are included in a single group (under Windows). Pietrek teaches describing modules related to a task with a task dependency list, and thus the combined teaching of Prosise and Pietrek would have provided the Windows and DOS applications with task dependency lists. Tulpule teaches partitioning tasks based on interdependencies, including examining task dependency lists (task's prerequisites, col. 4, lines 19-22; col. 6, lines 6-13). One of ordinary skill in the art would have been motivated to combine the teachings of Prosise as modified with Tulpule

because this would have provided a flexible configuration to dynamically respond to changes of task's execution times (Tulpule, col. 2, lines 29-34).

As to claims 6 and 7, note discussions of claims 2 and 3, respectively.

As to claim 14, note discussion of claim 5. Further, Prosise and Pietrek teaches operating system (Windows 3.x, Windows 3.1), system resources (Windows system routines, DOS system routines). Prosise and Pietrek teaches for each task, modules on the task dependency list for the task are included in a single group in that Windows OS and DOS each provides its own system routines, ie, resources, to be allocated/called by application programs running thereon. Windows applications grouped under System VM allocate/utilize Windows system routines / modules, and DOS applications grouped under DOS VMs allocate/utilize DOS system routines / modules.

6. Claims 8-10 and 15 are allowed.

7. Claims 4, 13 are objected to as being dependent upon a respective rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the respective base claim and any intervening claims.

8. Applicant's arguments filed 6/28/2004 have been considered but are moot in view of the new ground(s) of rejection.

Regarding claim 1, applicant argued that it would not be obvious to add cooperative / non-preemptive scheduling to DOS VM (remarks, page 9, 2nd paragraph and page 12, 1st paragraph).

The examiner's response is that Notes (Lecture Notes "Non-Preemptive Scheduling") is now cited to explicitly teach that both DOS and Windows 3.1 separately employ non-preemptively scheduling. See discussion of claim 1 for detail.

Applicant further argued that Tulpule does not teach grouping of tasks based on their interdependencies so that those groups may be preemptively multitasked (remarks, page 10, 2nd paragraph).

The examiner's response is that it is the combination of Prosise, Pietrek and Tulpule, rather than Tulpule alone, that meets the claimed limitation, as detailed in the rejection of claim 5:

'As to claim 5, it is covered by claim 1 except for (1) modules of code, (2) providing a task dependency list for each task, the task dependency list listing resources that are candidates to be allocated when the task is run, (3) that the logically partitioning includes examining the task dependency lists to group the tasks into groups of interdependent tasks.

As to (1)-(2), Pietrek teaches

As to (3), Prosise teaches interdependent tasks (windows applications) are grouped together (grouped to be run under System VM), wherein the modules run by the tasks (windows resources, windows DLLs) are included in a single group (under Windows). Pietrek teaches describing modules related to a task with a task dependency list, and thus the combined teaching of Prosise and Pietrek would have provided the Windows and DOS applications with task dependency lists. Tulpule teaches partitioning tasks based on interdependencies, including examining task dependency lists (task's prerequisites, col. 4, lines 19-22; col. 6, lines 6-13). One of ordinary skill in the art would have been motivated to combine the teachings of Prosise as modified with Tulpule because this would have provided a flexible configuration to dynamically respond to changes of task's execution times (Tulpule, col. 2, lines 29-34).'

Applicant argued that not all the teachings of Prosise, Pietrek was necessarily available in 1992 because Windows 3.11 was available later than 1992 (remarks, paragraph bridging pages 10 and 11).

The examiner's response is that while Windows 3.11 might be available later than 1992, applicant has not provided evidence showing that the teachings of Prosise and of Pietrek relied on was only available in Windows 3.11 and/or later versions and not available in Windows 3.1 and/or earlier versions. Therefore, the argument is not persuasive. Prosise discusses features of Windows 3.1 and Windows 3.x which includes Windows 3.1. Stinson shows that Windows 3.1 was available to the public

before the filing date of the present application. Therefore, the teaching of Prosise was available before the present filing date.

Regarding claims 2-3, 6-7, applicant argued that "[t]he TDB is not a group list as recited in Claim 6, but rather a series of tasks that happen to be linked together through the use of the selectors" "the selectors of Pietrek are not information that is identified for the tasks, but merely identifiers of the tasks themselves." (remarks, pages 12, 14-15).

The examiner's response is that claim 2 requires "a group list for each associated group in the storage device, wherein each group list includes identifying information for tasks included in the associated group." Claim 3 "identifying information about any task that is running". Clearly, no specifics are recited to preclude the application of Pietrek to meet the claimed a group list for each associated group (task database TDB, maintained in a linked list), identifying information for tasks (selector of next TDB), storing status information (flags, error flags, etc.), whether the group has a task that is running (DLL is loading, DLL is unloading), holding identifying information about any task that is running (module handle for task). It is noted that 'identifying information' is very broad and is clearly met by the task selectors and task handles of Pietrek.

Regarding claims 5, 14, applicant argued that "modules run by Windows applications, such as Windows resources and DLLS, could be considered part of a Windows group is not the same as interdependent tasks that are grouped together because of their interdependency" (remarks, pages 13-14, 15).

The examiner's response is that Windows OS and DOS OS each provides its own system routines, ie, resources, to be utilized/called by application programs running thereon. Windows applications grouped under System VM call Windows' system routines and resources, and similarly DOS applications grouped under DOS VMs call DOS system routines and resources. In other words, Windows applications depend on windows' resources to run, ie, Windows applications and windows' resources have a interdependency relationship among themselves. The same is true for DOS applications and DOS resources.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (571) 272-3764. A voice mail service is also available at this number. The examiner's supervisor, SPE Meng-Ai An, can be reached on (571) 272 3756. The examiner can normally be reached on Monday - Friday, from 9AM to 5PM. The fax phone number for the organization where this application or proceeding is assigned is (703) 872 9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

November 22, 2004



SUE LAO
PRIMARY EXAMINER